

LISTING OF THE CLAIMS

Claim 1 (Currently Amended): A digital rights management system for controlling the distribution of digital content to player applications including a core rendering application and an extension mechanism, the system comprising:

a verification system to validate the integrity of the player applications, and including a certificate generator for generating a certificate after inspecting the player application code and determining that a certain required property has been met by said code;

a trusted content handler to decrypt content and to transmit the decrypted content to the player applications, using an the extension mechanism ~~defined by~~ of the player application, and to enforce usage rights associated with the content; and

a user interface control module to ensure that the user interaction with the player applications does not violate the usage rights by intercepting and filtering messages sent from the user to the player application in accordance with a user rights set obtained by the user;

wherein components of the verification system, the trusted content handler, and user interface control module of the digital rights management system operate independently from the player application, reside locally in an end-user device having said player applications, and are dynamically linked to the applications at run-time; and

wherein the digital rights management system uses the extension mechanism of the player applications to implement the functions of the digital rights management system without modifying the core rendering applications of the player applications.

Claim 2 (Original): A digital rights management system according to Claim 1, wherein the verification system includes an off-line verifier to verify that the player applications have certain properties, and to issue trust certificates to verify that the player applications have said properties.

Claim 3 (Original): A digital rights management system according to Claim 2, wherein the verification system further includes a verifying launcher for verifying that a particular player application is certified as a trusted application before digital content is transmitted to said particular player application.

Claim 4 (Original): A digital rights management system according to Claim 1, wherein the player applications request protected content, and the trusted content handler includes an authenticator to verify that a player application that requests protected content has been authorized by the verification system to access the requested, protected content.

Claim 5 (Original): A digital rights management system according to Claim 1, wherein a user interface control module traps user interface related messages generated as a result of user interactions with player applications, blocks messages that lead to usage rights violations, and passes through other messages to the player applications.

Claim 6 (Currently Amended): A digital rights management method for controlling the distribution of digital content to player applications including a core rendering application and an extension mechanism, the method comprising the steps:

providing a verification system to validate the integrity of the player applications, said verification system including a certificate generator for generating a certificate after inspecting the player application code and determining that a certain required property has been met by said code;

using a trusted content handler to decrypt content and to transmit the decrypted content to the player applications, using ~~an~~ the extension mechanism ~~defined by~~ of the player applications, and to enforce usage rights associated with the content; and

providing a user interface control module to ensure that the user interaction with player applications does not violate the usage rights by intercepting and filtering messages sent from the user to the player application in accordance with a user rights set obtained by the user;

wherein components of the verification system, the trusted content handler, and user interface control module of the digital rights management system operate independently from the player application, ~~and~~ reside locally in an end-user device having said player applications, and are dynamically linked to the application at run-time; and

wherein the digital rights management system uses the extension mechanism of the player applications to implement the functions of the digital rights management system without modifying the core rendering applications of the player applications.

Claim 7 (Original): A method according to Claim 6, wherein the step of providing a verification system includes the step of providing an off-line verifier to verify that the player applications have certain properties, and to issue trust certificates to verify that the player applications have said properties.

Claim 8 (Original): A method according to Claim 7, wherein the step of providing a verification system further includes the step of providing a verifying launcher for verifying that a particular player application is certified as a trusted application before digital content is transmitted to said particular player application.

Claim 9 (Original): A method according to Claim 6, wherein the player applications request protected content, and the step of using the trusted content handler includes the step of using an authenticator to verify that a player application that requests protected content has been authorized by the verification system to access the requested, protected content.

Claim 10 (Currently Amended): A program storage device readable by machine, tangibly embodying a program of instructions executable by the machine to perform method for controlling the distribution of digital content to player applications including a core rendering application and an extension mechanism, the method steps comprising:

using a verification system to validate the integrity of the player applications, said verification system including a certificate generator for generating a certificate after inspecting the player application code and determining that a certain required property has been met by said code;

using a trusted content handler to decrypt content and to transmit the decrypted content to the player applications, using an the extension mechanism defined by of the player applications, and to enforce usage rights associated with the content by intercepting and filtering messages sent from the user to the player application in accordance with a user rights set obtained by the user; and

using a user interface control module to ensure that the user interaction with player applications does not violate the usage rights;

wherein components of said verification system, the trusted content handler, and user interface control module operate independently from the player applications and reside locally in an end-user device having said player applications, and are dynamically linked to the applications at run-time; and

wherein the digital rights management system uses the extension mechanism of the player applications to implement the functions of the digital rights management system without modifying the core rendering applications of the player applications.

Claim 11 (Original): A program storage device according to Claim 10, wherein the step of using the verification system includes the step of using an off-line verifier to verify that the player applications have certain properties, and to issue trust certificates to verify that the player applications have said properties.

Claim 12 (Original): A program storage device according to Claim 11, wherein the step of using the verification system further includes the step of using a verifying launcher for verifying that a particular player application is certified as a trusted application before digital content is transmitted to said particular player application.

Claim 13 (Original): A program storage device according to Claim 10, wherein the player applications request protected content, and the step of using the trusted content handler includes the step of using an authenticator to verify that a player application that requests

protected content has been authorized by the verification system to access the requested, protected content.

Claim 14 (Currently Amended): A code identity and integrity verification system for verifying the integrity or code of player applications including a core rendering application and an extension mechanism, comprising:

a certificate generator for receiving the player applications, for inspecting the player applications code to determine if the player applications code exhibit a predefined property, and for issuing a trust certificate for each of the player applications that exhibits the predefined property;

a certificate repository for receiving and storing trust certificates issued by the certificate generator;

an off-line code verifier for to analyze program code of a particular player application to determine whether said particular player application is certified as a trusted application before digital content is transmitted to said particular player application; and

an authenticator for receiving requests, using an the extension mechanism defined by of one of the player applications, to verify that [[a]] said one player application that requests protected content has been authorized by the verification system to access the requested, protected content, wherein the authenticator operates independently from said applications, resides locally in an end-user device having said applications, and is dynamically linked to said applications at run-time; and

wherein the authentication uses the extension mechanism of said one player application to implement the functions of the authenticator without modifying the core rendering application of said one player application.

Claim 15 (Original): A code identify and integrity verification system according to Claim 14, wherein the code verifier is responsible for launching the player application and verifying the identity and integrity of the code using the information in the trust certificate before launching the application; the launch procedure returning process identification information, which the code verifier records internally; the authenticator communicating the same or other process identification information concerning its own process, which it obtains from system service calls, to the code verifier at the time the application requests content from the authenticator; the code verifier matching this process identification information against the process identification information it recorded; the code verifier returning a code indicating whether the process was verified or not.

Claim 16 (Original): A code identity and integrity verification system according to Claim 14, wherein the code verifier receives from the authenticator process identification information at the time the player application calls the authenticator; the code verifier querying the operating system with the process identification information or the file names of all modules loaded for that process; the code verifier using the information in the trust certificate to verify the identity and integrity of the code modules; returning a code indicating whether the process was verified or not.

Claim 17 (Original): A code identity and integrity verification system according to Claim 14, wherein the trust certificate includes:

- a program identifier identifying said one of the applications;
- a property name identifying an attribute certified by the trust certificate;
- a code digest of the one application;

a digital signature containing a secret key of the application certifier; and
a certifier identification containing a public key of the application certifier.

Claim 18 (Currently Amended): A method for verifying the identity and integrity of code of player applications including a core rendering application and an extension mechanism, the method comprising the steps:

using a certificate generator for receiving the player applications, for inspecting the player applications code to determine if the player applications code exhibit a predefined property, and for issuing a trust certificate for each of the player applications that exhibits the predefined property;

receiving and storing in a certificate repository trust certificates issued by the certificate generator;

using an off-line code verifier to analyze program code of a particular player application to determine whether said particular player application is certified as a trusted application before digital content is transmitted to said particular player application; and

using an authenticator for receiving requests, using ~~an~~ the extension mechanism ~~defined by~~ of one of the player application, to verify that [[a]] said one player application that requests protected content has been authorized by the verification system to access the requested, protected content, wherein the authenticator operates independently from said applications, resides locally in an end-user device having said applications, and is dynamically linked to said applications at run-time; and

wherein the authentication uses the extension mechanism of said one player application to implement the functions of the authenticator without modifying the core rendering application of said one player application.

Claim 19 (Previously Presented): A method according to Claim 18, wherein the trust certificate includes:

- a program identifier identifying said one of the applications;
- a property name identifying an attribute certified by the trust certificate;
- a code digest of the one application;
- a digital signature containing a secret key of the application certifier; and
- a certifier identification containing a public key of the application certifier.

Claim 20 (Currently Amended): A program storage device readable by machine, tangibly embodying a program of instructions executable by the machine to perform method steps for verifying, out of process, the identity of code of player applications including a core rendering application and an extension mechanism, said method steps comprising:

- using a certificate generator for receiving the player applications, for determining if the player applications exhibit a predefined property, and for issuing a trust certificate for each of the player applications that exhibits the predefined property;

- receiving and storing in a certificate repository trust certificates issued by the certificate generator;

- using an off-line code verifier to analyze program code of a particular player application to determine whether said particular player application is certified as a trusted application before digital content is transmitted to said particular player application; and

- using an authenticator for receiving requests, using an the extension mechanism defined by of one of the player application, to verify that [[a]] said one player application that requests protected content has been authorized by the verification system to access the requested, protected content, wherein the authenticator operates independently from said

applications, resides locally in an end-user device having said applications, and is dynamically linked to said applications at run-time; and

wherein the authentication uses the extension mechanism of said one player application to implement the functions of the authenticator without modifying the core rendering application of said one player application..

Claim 21 (Cancelled).

Claim 22 (New): A method according to Claim 6, for controlling the distribution of digital content to the player applications when said player applications are installed on end user devices, and wherein:

the step of providing a verification system includes the step of employing the verification system to use a specified verification process to validate the integrity of the player applications and without requiring the player applications to participate in the verification process while the player applications are installed on the end user devices; and

comprising the further steps of:

one of the end users generating a public key/private key pair, registering with a digital rights management server, and sending the public key to said server;

said server using said public key to encrypt a rights file and a content map, and then sending the encrypted rights file and content map to said one of the end users; and

said one of the end users using the private key to decrypt said encrypted rights file and content map.